Code and Data
for arXiv:2208.09983

Guang Ping He
School of Physics, Sun Yat-sen University, Guangzhou 510275, China
hegp@mail.sysu.edu.cn

Overview:

All raw data of the experiments in the paper are in the file “experimental_data.zip”.

All the software code of the Python programs for generating these data are located in the “programs” directory.

About the programs:

The MNIST data set and *mnist_loader.py* (not included here) will be called by all programs except *weight_extractor.py*. Available at:
https://github.com/mnielsen/neural-networks-and-deep-learning/archive/master.zip

All programs with the names ending with “S” are for the networks using sigmoid function.

- *main2204_2Para6S.py*:
  A batch processing file that completes all the training and analysis (except the function of *weight_extractor.py*) of a network.
  It includes *main2204_2Para_load_6S.py*.
  It will call *network2Para6S.py* and *network2Para_load_6S.py*.
  It will produce the following files:
  biases&weights_initialized.pickle
  biases&weights_optimized.pickle
  network2Para6S.pkl
  classification_accuracy.csv
  disagreed_results.csv
  Main parameters that can be changed:
  [784,48,35,10],[784,50,10]: the size of the networks.
  epochs in the function *SGD*: the number of epochs that the networks are trained in total.
  epochs_sep in the function *SGD*: the number of epochs that the networks are trained separately.
- *main2204_2Para_load_6S.py*:
  A batch processing file that analyzes a trained network.
  It will call *network2Para_load_6S.py*.
  It will read *biases&weights_optimized.pickle*, and produce *disagreed_results.csv*.
- *network2Para6S.py*:

The program that trains a network.
It will produce the following files:

biases&weights_initialized.pickle
biases&weights_optimized.pickle
network2Para6S.pkl
classification_accuracy.csv

- *network2Para_load_6S.py*:
  The program that analyzes a trained network.
  It will read *biases&weights_optimized.pickle*, and produce *disagreed_results.csv*.

All programs with the names ending with “R” are for the networks using ReLU function.

All programs with the names ending with “T” are for the networks using Tanh function.

- *weight_extractor.py*:
  The program that extracts the weights of the outputs of the neurons of the last hidden layer when calculating the weighted inputs to the output neurons.
  It will read *biases&weights_optimized.pickle* and produce *weights_at_the_output_layer.csv*.

Experimental data file description:

- *biases&weights_initialized.pickle*: The initial biases and weights of the network.

- *biases&weights_optimized.pickle*: The optimized biases and weights of the network.

- *classification_accuracy.csv*: The classification accuracies at each epochs and their maximums.
  Notations (left: symbols in the file, right: symbols in the paper):

  NN1: α_1
  NN1p: α'_1
  NN2: α_2
  NN2p: α'_2
  Para: α_{para}

- *disagreed_results.csv*: The analysis result of a trained network.
  Notations (left: symbols/phrases in the file, right: symbols/phrases in the paper):

  NN1: α_1
  NN1p: α'_1
  NN2: α_2
  NN2p: α'_2
  Para: α_{para}
  Results when both subnetworks are right: Type I results
  Results when only network1 is wrong: Type II results

Results when only network2 is wrong: Type III results

Results when both subnetworks are wrong but the combined network is right: Type IV results

- *network2Para6S.pkl* / *network2Para6R.pkl* / *network2Para6T.pkl*: The network at the end of the training.

- *weights_at_the_output_layer.csv*: The weights of the outputs of the neurons of the last hidden layer when calculating the weighted inputs to the output neurons.